A Hybrid Intrusion Detection System for Virtual Jamming Attacks on Wireless Networks

Diego Santoro, Ginés Escudero-Andreu, Konstantinos G. Kyriakopoulos, Francisco J. Aparicio-Navarro, David J. Parish, and Michele Vadursi

***Abstract*—Wireless communications are vulnerable to certain number of cyber-attacks and intrusion attempts due to the intrinsic openness of the communication channel. Virtual jamming attack stands out among other attacks. This type of attack is easy to implement, energy-efficient to be launched, and represents one of the most important threats to the security of wireless networks. As the complexity of the attacks keeps increasing, new and more robust detection mechanisms need to be developed. A number of Network Intrusion Detection Systems (NIDSs) have been presented in the literature to detect this type of attack. To tackle the problem of virtual jamming attacks on IEEE 802.11 networks, we present a novel Hybrid-NIDS (H-NIDS) based on Dempster-Shafer (DS) Theory of Evidence. The proposed method aims at combining the advantages of signature-based and anomaly-based NIDSs. The performance of the proposed solution has been experimentally evaluated with multiple scenarios in an IEEE 802.11 network.**

***Index Terms*—Data Fusion, Intrusion Detection Systems, Measurements and Networking, Network Security, Virtual Jamming Attacks, Wireless Network Measurements**

I. INTRODUCTION

THE intrinsic physical openness of wireless communication channels exposes wireless and cellular networks to a certain number of attacks, such as jamming that can be very difficult to trace [1, 2]. Today, jamming attacks are rather easy to implement, considering a number of off-the-shelf tools is available [3-5], and thus represent one of the most serious threats to the availability of wireless networks. Jamming attacks can be classified as physical jamming and virtual jamming. Examples of the former are: radio jamming, where the attacker continuously transmits a radio signal carrying random bits, and collision attack, where the attacker sends a packet only when it senses that a legitimate user is sending a valid packet, so as to cause a collision [6]. Examples of virtual jamming are: spurious Request-To-Sent/Clear-To-Sent (RTS/CTS) attacks, which consist in sending fake RTS frames, and Network Allocator Vector (NAV) attacks, where the attacker alters the duration field of legitimate packets. Both types of attack aim to delay the transmission of legal frames. Compared to physical jamming, virtual jamming is easier to implement and needs little power to be carried out.

A number of Network Intrusion Detection Systems (NIDSs) have been presented in the literature to detect a wide range of jamming threats [7-12]. A NIDS can be classified as a signature-based NIDS (also known as misuse-based NIDS) or an anomaly-based NIDS. The former detects attacks by comparing the network traffic profile with signatures of well-known threats or attacks. This type of NIDS is generally very efficient and accurate, but fails to identify attacks that do not belong to the set of reference signature. This includes attacks that are launched for the first time or attacks that slightly differ from their former and known implementations. On the other hand, an anomaly-based NIDS compares the network traffic profile against a baseline representing the normal (attack-free) behaviour of the network. Generally, this type of NIDS is not as accurate as signature-based NIDSs, since its performance is poorer in terms of False Positive Ratio (FPR) [13]. However, unlike signature-based NIDSs, anomaly-based NIDSs can successfully detect novel and unseen attacks and threats. More recently, Hybrid-NIDSs (H-NIDSs) have been developed to combine the detection capabilities and the aforementioned advantages of both types of NIDSs: the high Detection Rate (DR) of signature-based NIDSs with the ability to detect novel attacks of anomaly-based NIDSs [14-16].

The authors have recently presented an anomaly-based NIDS to detect virtual jamming NAV attacks on IEEE 802.11 networks [17]. The core detector adopts an anomaly-based approach, which exploits an implementation of the Dempster-Shafer (DS) Theory of Evidence [18]. The performance of the detection algorithm is overall encouraging, but it suffers from high FPR and False Negative Ratio (FNR) when certain combinations of metrics are used.

In this paper, we present a novel H-NIDS to detect virtual

Manuscript originally received April 28, 2016. Revised on March 24, 2017. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014307/2 and the MOD University Defence Research Collaboration in Signal Processing.

G. Escudero-Andreu, K. G. Kyriakopoulos, and D. J. Parish are with the Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough, LE11 3TU, UK (e-mail: {elge2, elkk, d.j.parish}@lboro.ac.uk). K. G. Kyriakopoulos is also with the Institute for Digital Technologies, Loughborough University London, E20 3BS, UK.

F. J. Aparicio-Navarro is with the School of Electrical and Electronic Engineering, Newcastle University, Newcastle upon Tyne, NE1 7RU, UK (e-mail: francisco.aparicio-navarro@ncl.ac.uk).

D. Santoro, and M. Vadursi are with the Department of Engineering University of Naples "Parthenope", Napoli, Italy (e-mail: {diego.santoro; michele.vadursi}@uniparthenope.it).

jamming attacks on IEEE 802.11 networks. This H-NIDS extends and improves the method that we previously proposed in [17], with a completely rearranged architecture. With this proposed hybrid approach, we want to improve the detection accuracy of the system, and reduce the number of false alarms.

Our contribution in this paper can be summarised as follows:

First, we propose a novel framework for an H-NIDS to combine the detection capabilities and the advantages of two types of NIDSs. In particular, the core detector architecture of the proposed solution runs in parallel an anomaly-based and a signature-based detection engine. Then, the detector applies the Dempster's rule of combination on the two independent pieces of information.

Second, the performance of the novel hybrid solution is evaluated against virtual jamming attacks in an IEEE 802.11 network environment. A new network traffic dataset has been collected for evaluation purposes, which takes into account new relevant scenarios not previously studied in [10]. Additionally, in this work we consider a wider set of metrics not previously used in [10], which could manifest the presence of an attack. The selection of the metrics has been done experimentally, during a pre-processing stage.

Finally, the whole solution is implemented as a single monitoring station, which derives the metrics by observing the current traffic within the IEEE 802.11 testbed network. Unlike other solutions proposed in the literature, the H-NIDS that we propose is implemented as a light, centralised and on-line solution. The architecture of the proposed detector allows the implementation of the detection process, with a reduced set of metrics, which induces a limited computational processing increase.

The paper is organised as follows. Section II presents an analysis of the state of the art related to jamming attack detection in Wi-Fi networks. The virtual jamming NAV attack and the proposed detection methodology, as well as the architecture of the H-NIDS are all described in Section III. The performance assessment is included in Section IV, which describes the implementation of the attack and the testbed. The analysis of the experimental results is presented in Section V. Finally, conclusions are given in Section VI.

II. RELATED WORK

Jamming attacks have been widely investigated in the literature. Multiple authors have proposed several solutions to tackle the problem for a wide range of jamming attacks [7-12]. In [7], the authors present DOMINO, a piece of software installed in or near an Access Point (AP) in order to detect MAC layer greedy behaviour in 802.11 hotspots. DOMINO is organised in three modules: (i) Deviation Estimation Component (DEC), (ii) Anomaly Detection Component (ADC) and (iii) Decision Making Component (DMC). The DEC module performs the following tests: retransmission consistency, DIFS consistency, NAV consistency and back-off manipulation test. DOMINO runs the tests for each node by tracking the transmission of each node in the network. Therefore, the required processing time and the computational cost of the analysis may become very demanding as the number of nodes in the network increases [8]. The performance of DOMINO was assessed using the network simulator NS-3 [19]. The results show that DOMINO is characterised by high detection accuracy and resiliency to several factors, such as traffic type variations.

The authors of [9] present a different solution, based on a distributed cross-layer detection system for a wide range of jamming attacks. The monitoring functionality is randomly distributed among the nodes, and the detection mechanism is organised in two phases. In the first phase, the system performs four tests on: (i) the physical idle time, (ii) the average number of RTS/CTS frames transmitted by a node, (iii) the virtual idle time (NAV), and (iv) the average number of retransmissions of a node. In the second phase, the results are combined and then a final test is carried out in order to increase accuracy. It is worth noting that this solution becomes time and resource consuming when the number of nodes in the network increases, since tests need to be carried out for each node. The performance is assessed through the simulator GloMoSim [20]. The results show that as the number of nodes increases, the data rate detection decreases, and the number of false positive increases.

In [17], an off-line detection algorithm is proposed, which is able to detect and classify physical and virtual jamming attacks. The algorithm needs the following metrics as inputs: Packet Delivery Ratio (PDR) and Packets Send Ratio (PSR). The algorithm outcomes are compared with a Signal Strength Consistency check in order to improve the overall system accuracy. The consistency test is necessary because, as the authors suggest, a low PDR might be caused by a node running out of battery or a user moving away from the coverage area. The used metrics have to be calculated for each node, and data have to be retrieved from transmitting and receiving nodes during the jamming attack. The simulation results show that the algorithm is characterised by high accuracy and precision rates.

A threshold-based NIDS to detect virtual jamming attacks on IEEE 802.11 networks was presented by the authors in [10], who take the decision on packets sent and delivery ratio. Similarly, against jamming attack, the authors in [11] propose a distributed solution to detect jamming attacks using only metrics from the physical layer. In detail, the method is based on the detection of changes in the statistical characteristics of the Signal-to-Noise Ratio (SNR). The detection is carried out locally by using either a simple-threshold algorithm or a CUSUM-based algorithm. An improved version of the method, based on DS theory that combines distributed sensor beliefs is also compared against local based algorithms. Prior work in DS theory also includes multi-metric, cross-layer anomaly-based techniques that have been evaluated for detection of Man-in-the-Middle (MitM) and de-authentication attacks [21].

More recently, H-NIDSs have been developed to combine the detection capabilities and the advantages of both types of NIDSs. An H-NIDS is proposed in [22] to detect attacks in a cloud computing environment. This H-NIDS implements a Bayesian classifier for the anomaly part and a SNORT [23] script for the misuse part of the core detector. The authors show that the solution is characterised by high DR and low FPR.

Another H-NIDS is proposed in [24] to detect Distributed Denial of Service attacks (DDoS) at the application layer, which implements a Bayesian classifier for the anomaly part and a Hidden Markov Model (HMM) for the misuse part of the detector. Similarly to [22], the detectors show high DR along with high FPR. Equally good results in terms of both DR and FPR are experienced in [25], where an H-NIDS based on Principal Component Analysis (PCA) and Self Organising Map (SOM) is presented.

In this paper, we propose a novel H-NIDS designed to detect virtual jamming attacks. Such type of attacks leverage the ability of an attacker to manipulate the NAV value, which is a prominent virtual carrier sensing mechanism in CSMA/CD. It is therefore a common characteristic of all 802.11 MAC based wireless networks and even used in other technologies, such as, WiMax [34, 35]. Furthermore, it is worth noting that, 4G/5G networks heavily rely on IEEE 802.11 as a Radio Access Technology (RAT) [26] for reducing the traffic overload and coping with a high and dynamic user density while sharing a finite radio spectrum. Therefore, addressing this particular attack in our scenario, is also applicable and beneficial towards mobile communications.

The core detector of the presented solution exploits the Dempster's rule of combination of DS to merge pieces of evidence of a possible attack. The information is inferred by using an anomaly-based and signature-based detection approach. Unlike the solutions listed previously, the H-NIDS that we propose is implemented as a light, centralised and on-line solution. The whole solution is implemented as a single monitoring station, which derives the metrics by observing the current traffic within the IEEE 802.11 testbed network.

III. PROPOSED DETECTION METHODOLOGY

A. Data Fusion Approach Based on DS Theory

The proposed H-NIDS is based on the use of evidence theory. In recent years, the theory of belief functions, also known as the theory of evidence developed by Dempster and Shafer [18], has drawn the attention of many researchers, especially in the fields of sensor and data fusion [27]. The DS theory provides a simple but robust framework to merge information coming from different sensors, taking into account the available pieces of evidence. In contrast to Bayesian theory, the DS theory does not require a priori knowledge and enables a way of measuring ignorance, when the evaluated data cannot be allocated within the considered hypotheses. It has proven to be a viable solution in cases where it is impossible to apply classical sensor fusion techniques, such as Kalman filter or Bayesian networks, or when it is virtually impossible to find a pattern in the system behaviour to build an appropriate model [11]. In addition, the DS theory has also been used to develop a new mathematical framework [27-29] alternative to the Guide to the Expression of Uncertainty in Measurement (GUM) [30] for the evaluation of the uncertainty in complex measurement systems.

The DS theory considers a set of events $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$, which is a finite set of all possible mutually exclusive propositions about some problem domain, known as frame of discernment. Regarding this work, the aim is to identify whether the analysed network traffic is malicious or non-malicious. Therefore, Θ is composed of two elements A = *Attack* and N = *Normal*. Assuming Θ has two outcomes $\{A, N\}$, the total number of hypotheses is defined by $2^{\Theta} = \{A, N, \{A|N\}, \emptyset\}$. In the case of $\{A|N\}$, this subset corresponds to *Uncertainty* (either A or N). In addition, the empty set $\emptyset$ is always null. Each hypothesis is assigned a belief value within the range [0, 1], also known as a Basic Probability Assignment (BPA), which expresses the evidence attributed directly to the hypothesis. The BPA is a function $m(H)$, which describes the measure of belief committed directly to the hypothesis H by an observer. It is worth noting that, in contrast to probability theory, the DS theory does not comply with the additivity rule [31].

After defining the BPA value for each hypothesis, the DS theory combines evidence of information from different observers or sensors with similar Θ using the Dempster's rule of combination [18]. This rule is defined in (1), and calculates the orthogonal summation of the BPAs values in one hypothesis from two different observers into a single belief. Let $m_1(H)$ and $m_2(H)$ be the BPA in the hypothesis H, from observer 1 and 2, respectively. Similarly, $X \cap Y = H$ refers to all combinations of evidence which yield H; whereas $X \cap Y = \emptyset$ refers to the mutually exclusive subsets of the hypothesis H, thus their intersection is the empty set.

$$m_{comb}(H) = \frac{\sum_{X\cap Y=H} m_1(X) * m_2(Y)}{1 - \sum_{X\cap Y=\emptyset} m_1(X) * m_2(Y)} \quad \forall\, H \neq \emptyset \quad (1)$$

Dempster's rule allows the combination of evidence from two observers at a time. In order to combine evidence from more observers, Dempster's rule can be used repeatedly several times in consecutive iterations. The output of the initial combination process is used as input evidence in the next iteration, along with the evidence of information from a third observer. Dempster's rule satisfices the associative property, thus the order in which the belief values are fused does not affect the final combined belief values. A more comprehensive presentation of DS theory is presented in [18].

An important issue affecting the development phase of a detector based on the use of evidence theory is how to define the BPA values. In the literature, there exist several ways of assigning probabilities to each of the hypotheses, ranging from data mining techniques to empirical approaches. One method to find an automatic and self-adaptive process of BPA without a previous training process or fine tuning period was initially presented in [21], using three independent statistical mechanisms.

B. Architecture of the Proposed Hybrid NIDS

The architecture of the proposed H-NIDS is shown in Fig. 1. It consists of three main blocks, the BPA function calculation block, enclosed in the dashed-borders box in Fig. 1, the data fusion block and the decision-making block. The BPA function calculation block reads the fields in the network frames, extracts the relevant monitored metrics and calculates the BPA values for each monitored metric and for each of the three considered hypotheses (i.e. *Attack*, *Normal* and *Uncertainty*).

The relevant metrics extracted for our experiments are described in Section IV.C. The data fusion block merges the computed BPA values for each metric and calculates the overall BPA values. Lastly, the decision-making block makes a final decision on whether a NAV attack is taking place or not, based on the final BPA values of the three considered hypotheses. Each of the three blocks is explained in more detail in the following subsections.

1) BPA Function Calculation Block

Regarding the first block, it contains a BPA calculator sub-block for each of the monitored metrics. Each sub-block contains two independent buffers: the anomaly-based buffer and the misuse-based buffer. The algorithm has an initial phase, where it gathers a number of incoming frames to fill the anomaly-based buffer. The anomaly-based buffer contains the metric's values that define the behaviour of network traffic without classified attacks and it is implemented as a FIFO queue of prefixed size. In contrast, the signatures in the misuse-based buffers are taken from previous attacks and are not dynamically updated. Specifically, to construct the misuse-based buffer at a prior stage, only attack traffic is passed as input to our BPA calculator, described below, which generates the actual attack signatures relating to each considered metric.

The metric BPA calculator block calculates the BPA value for the hypothesis of Attack ($m(A)$) and Normal ($m(N)$) by using the samples contained in both buffers. In more detail, the samples in both buffers are ordered in a low to high order, the percentiles (rather than quartiles as used in [21]) are calculated, and then the percentile within which the incoming metric's value falls into is equated to the BPA value. The BPA for the hypothesis Uncertainty ($m(U)$) is calculated as a correction factor using the methodology presented in [21].

During the course of the detection process, if the BPA values computed inside the metric BPA calculator matches the exact same BPA values computed on the misuse-based buffer for all the hypothesis (as seen in the "Equal?" condition in Fig.1), the metric value is discarded from being added to the anomaly-based buffer. The final decision of updating the anomaly-based buffer or discarding the metric value is taken on the basis of the result of a Boolean expression, which is true when the BPA value for the hypothesis normal ($m(N)$) is strictly greater than the BPA values of the other two remaining hypotheses. If the $m(N)$ for the incoming analysed metric value is the largest BPA of the three, then the metric value is included in the anomaly-based buffer and the FIFO queue is updated. The aim of this approach is to allow the anomaly-based buffers to dynamically adapt themselves to new operational conditions of the network and to improve the overall detector performance. The misuse-based part of the algorithm is used as feed-back loop into the hybrid detection algorithm.

2) Data Fusion Block

The second block, the data fusion block, merges the BPA values for each metric and calculates the overall BPA value for each hypothesis by using the DS rule of combination presented in Section III.A. The DS technique fuses the outcome of block 1, which was produced while considering anomaly-based and signature-based information, making the proposed detector a hybrid approach. Since DS can only merge two set of beliefs at a time, the data fusion block implements an iterative method when more than two metrics are considered.

3) Decision-Making Block

Finally, the third block, named decision-making block, is the one that makes the decision according to the outcome values of the BPA values. The hypothesis with the highest BPA value is considered the correct decision.

IV. EXPERIMENTAL FRAMEWORK

A. Virtual Jamming Attack Description

One example of a virtual jamming is the NAV attack. This is the attack that we have used in our experiments for this work. The NAV attack exploits the virtual carrier-sensing mechanism, a mechanism proposed in the IEEE 802.11 standard which aims to mitigate the collisions resulting from the hidden-terminal problem. Specifically, the header of each IEEE 802.11 packet contains a particular field, named duration, which determines in milliseconds the time needed to transmit the packet on the channel and the time interval during which the channel will be busy.

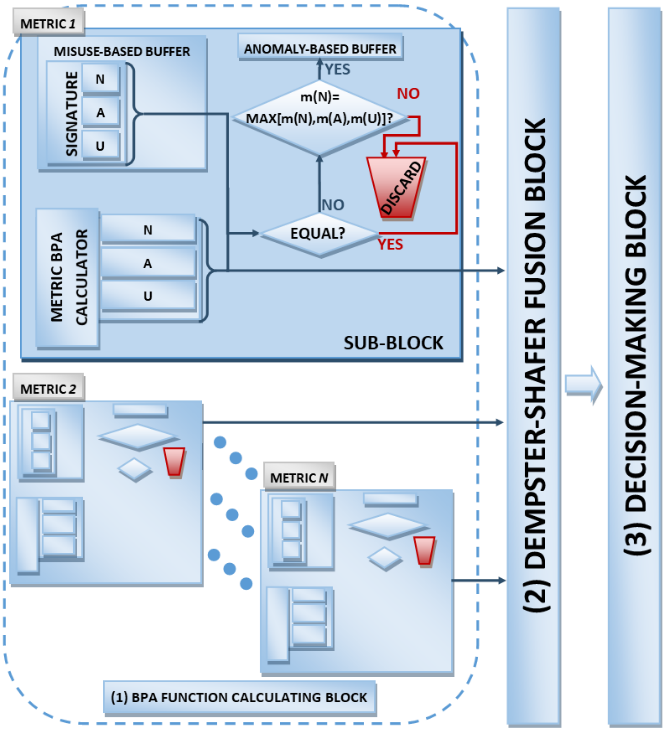

Figure 1. Architecture of the proposed hybrid NIDS.

As part of the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), every node in the wireless network reads the value of the duration field in order to set its own NAV timer. Assuming that the channel is busy and other node has something to transmit, the rest of nodes in the network will wait a period equal to NAV before start transmitting. After setting their NAV timer, the nodes start decreasing their back-off time. When the back-off timer reaches zero, if the channel is idle, then the nodes start transmitting; otherwise, they defer their transmission again. The overall CDMA/CA procedure is depicted in Fig. 2.

To carry out a NAV attack, the attacker overwrites two mechanisms of the IEEE 802.11 protocol: the RTS/CTS mechanism and the procedure to calculate the back-off time. Within the RTS/CTS mechanism, the field duration of each RTS packet is set by the attacker to the maximum NAV value 32767 (i.e. 32ms). Consequently, all nodes listening to the wireless channel will set their NAV timers to the maximum value and wait for the maximum time-interval to get access to the channel. On the other hand, the contention window of the back-off calculation mechanism is set to zero so that the attacker transmits in the very first idle time slot. Because the attacker is the first node to occupy the transmission channel, this attack makes all the wireless devices in the network to postpone any transmission.

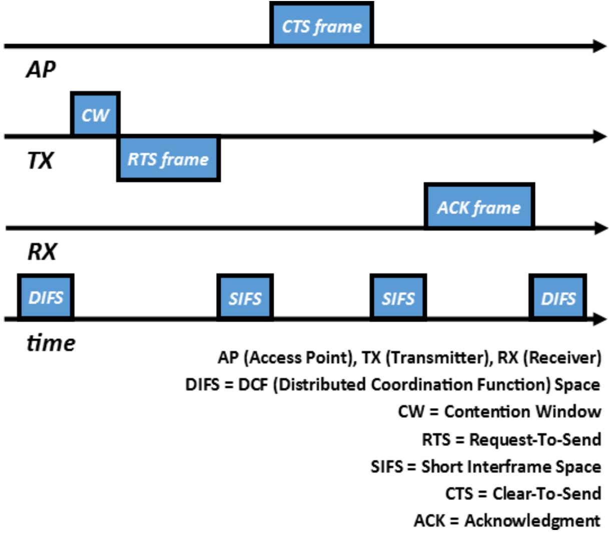

Figure 2. Representation of the virtual carrier-sensing mechanism.

B. IEEE 802.11 Network Testbed Description

The performance of the proposed H-NIDS has been evaluated on an experimental IEEE 802.11 network testbed, which was set up in our laboratory at the Wolfson School at Loughborough University. With this testbed, depicted in Fig. 3, we wanted to reproduce a realistic Wi-Fi scenario. The network testbed is composed of one AP and four nodes with different roles: Attacker, Monitor and two Clients.

The Attacker (or jammer) runs on Linux Ubuntu 10.04 Lucid Lynx. The wireless Network Interface Controller (NIC) is equipped with the Atheros 5100 chip, which is controlled by the ATH 5K driver. As explained in the previous subsection, this driver has been modified, to ignore the timeout imposed by the RTS/CTS mechanism and by fixing a static value for the collision window defined by the back-off mechanism. More specifically, the Atheros 5K has been modified to incorporate two changes in the desc.c and base.c files. The modifications, maximise the NAV value (i.e. set 'txctl2 |= AR5K_4W_TX_DESC_CTL2_RTS_DURATION') and disable the contention window (i.e. CWMIN and CWMAX are set to zero in the "ath5k_txq_setup()" function) as defined in the desc.c and base.c files, respectively. The driver is loaded as a new module in the kernel, forcing automatic binding with the hardware during the system initialisation.

The Monitor node also uses a NIC equipped with the Atheros 5100 chip. The NIC is configured in Monitor Mode to listen to the wireless channel. We used Wireshark [32] to collect the network traffic and our modified version of the ATH5000 driver to gather live statistics regarding the Cyclic Redundancy Check (CRC) error rate from the wireless interface card.

Lastly, we have used two Client nodes during our experiments, namely *Client_A* and *Client_B*. The clients follow the indications of the IEEE 802.11 standard, implementing the virtual carrier-sensing mechanism. Both clients send traffic during the whole monitoring period, and act as victims of the virtual jamming attack. The traffic generated by these nodes is artificially generated by using the Linux command iPerf [33] to send UDP and TCP traffic at a constant bit rate..

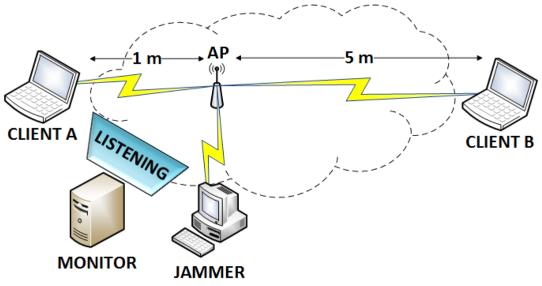

Figure 3. IEEE 802.11 Network Testbed Architecture.

The experimental campaign is summarised in Table I. In order to validate our solution, a list of 14 scenarios is proposed, including both nodes located in a static position throughout the entire traffic capturing phase, and mobile nodes constantly in movement. The first two scenarios (1 and 2 in Table I) are normal scenarios where the attacker is inactive. Only non-malicious network traffic is collected from these two scenarios. All the remaining scenarios (i.e. from 3 to 14) have a total duration of 90 seconds. Each scenario comprises three phases of the same duration: (i) initial phase, where only the well-behaved nodes send traffic, (ii) attack phase, where the attacker node initiates the virtual jamming attack, and (iii) final phase, where the attacker stops the attack.

TABLE I. EXPERIMENTAL SCENARIOS

Scenario	Description	RTS/CTS
1	No attacker, fixed ClientA and ClientB	-
2	No attacker, fixed ClientA and fiexed ClientB (with high NAV value)	-
3	Fixed ClientA	-
4	Fixed ClientA and ClientB	-
5	Moving ClientA	-
6	Moving ClientA and ClientB	-
7	Fixed ClientA and moving ClientB	-
8	Moving ClientA and fixed ClientB	-
9	Scenario 1 with RTS/CTS enabled	ClientA
10	Scenario 2 with RTS/CTS for single host	ClientB
11	Scenario 4 with RTS/CTS enabled	Both
12	Scenario 4 with RTS/CTS for single host	ClientB
13	Scenario 7 sending TCP traffic	ClientA
14	Fixed ClientB sending TCP traffic	ClientA

The test cases are designed to replicate realistic scenarios of movement activity in Wi-Fi networks. The clients are setup by following two main configurations; Firstly, in the fixed scenarios, *Client_A* is placed 1 meter away from the AP when acting as static or fixed node, while, secondly, *Client_B* is placed 5 meters away. Fixed nodes maintain a constant distance from the AP throughout each scenario for the purpose of keeping stable parameter values in the received radio signal. Random movements are also introduced in moving nodes' path to inflict signal variations, which has a direct impact on the bit rate. The movement reproduces a normal walking pace within an indoor environment, keeping a distance between 1 and 10 meters from the AP. Additionally, the combination of both fixed and moving nodes provides a more realistic assessment of the proposed detection algorithm when multiple clients are competing for the available radio resources, while being affected by the virtual jamming attack.

The devices in all these scenarios used UDP traffic. Only in scenario 14 were the effects of the virtual jamming attack on network traffic over TCP evaluated, which includes the establishment of a TCP session through the three-way handshake process.

C. Metrics Description

Multiple metrics were extracted from the network frames that compose the analysed datasets. The monitored metrics are: the NAV value, the inter-arrival time between consecutive frames (ΔT), the Frame Sequence Number (FSN) and the CRC errors.

As we have explained in Section IV.A, an attacker can modify the NAV value in the network frames to carry out a virtual jamming attack. Therefore, it is a sensible decision to use the NAV as part of the detection process. Monitoring the NAV value to detect a greedy behaviour or NAV attacks is common in the literature [7, 9, 12]. However, detecting intelligent jamming attacks by simply monitoring the NAV is not a robust solution because legitimate frames may carry high NAV values [34]. The ΔT is also monitored because this metric is generally affected during the virtual jamming attack. The main effect of a jamming attack is a service disruption, causing bandwidth reduction. In turn, the ΔT would increase during a jamming attack. Therefore, a virtual jamming attack may also manifest itself in an increase of the ΔT values. We have also taken into consideration the FSN metric, which has detectable peaks in the first order time differences of the frame sequence numbers, ΔFSN. The FSN metric presents these differences because the Wi-Fi card buffers overflow during the attack and it causes some network frames to be dropped. In [9] and [10], the authors describe that an increase in the number of damaged packets is observable during a virtual jamming attack. Both papers point out that in collision attacks, the number of CRC errors raises. This phenomenon is observable in our scenarios soon after the attack in launched. In fact, since the attacker sets its contention window to zero, it forcefully takes over the channel causing many collisions. Because of this reason, we have also added the CRC to the monitored metrics.

V. EXPERIMENTAL RESULTS

A. Performance Metrics

This section describes the detection results generated by the presented H-NIDS. The performance of the proposed solution has been evaluated using four well-known parameters, True Positive (TP), which represents malicious frames correctly classified as attacks; True Negative (TN), which represents normal frames correctly classified as non-malicious; False Positive (FP), which represents non-malicious frames misclassified as attacks; and False Negative (FN), which represents malicious frames misclassified as normal. These parameters are essential to calculate the following performance metrics, which provide quantifiable evidence of how effective the IDSs are at making correct detections.:

- Detection Rate (DR): Proportion of malicious frames correctly classified as attacks among all the malicious frames. DR (%) = $TP/(FN+TP)\cdot 100$
- False Positive Rate (FPR): Proportion of normal frames misclassified as malicious among all the frames. FPR (%) = $FP/(TP+TN)\cdot 100$
- False Negative Rate (FNR): Proportion of malicious frames misclassified as normal among all the malicious frames. FNR (%) = $FN/(FN+TP)\cdot 100$
- Overall Success Rate (OSR): Proportion of all the frames correctly classified.
  OSR (%) = $(TN+TP)/(TP+FP+TN+FN)\cdot 100$
- Precision: Proportion of malicious frames correctly classified as attacks among all the alarms generated.
  Precision (%) = $TP/(TP+FP)\cdot 100$
- F-Measure: Also known as *F-Score* and represents the weighted harmonic mean of Precision and DR.
  F-Score = $(2\cdot Precision\cdot DR)/(Precision + DR)$

B. Performance Analysis Under Normal Traffic

The H-NIDS has been initially evaluated in scenarios 1 and 2, when no attack takes place, to evaluate the performance of the proposed H-NIDS in terms of FPR. In scenario 1, both ClientA and ClientB transmit using a low NAV value. In scenario 2, ClientA also transmits using a low NAV value, whereas ClientB transmits using a high NAV value.

The experimental results for these two scenarios are reported in Table II, which shows the FPR results generated by each possible metric combination. The results show that the best single metric in terms of FPR for both scenarios is the NAV. Although the FPR for the NAV is around 2% in scenario 1, it exceeds 18% in scenario 2. Such drop in the performance is because in scenario 2, one of the legitimate users has set a high NAV value, which misleads the detection algorithm to produce a high number of FPs. In fact, the frames with the high NAV value are 17% of the total frames in the scenario 2 and all of them have been detected as malicious frames.

Regarding the multiple metric combinations, a noteworthy general improvement with FPRs in comparison to the results previously presented in [17] is made possible by the hybrid nature of the proposed NIDS.

The FPRs for all multi-metric combinations are below 15%, and all FPR results except for (ΔT, ΔFSN), (ΔT, CRC) and (ΔT, ΔFSN, CRC) are below 10%. In detail, we can observe improvement in the FPRs as more metrics are combined. For instance, in scenario 2, the FPR generated by the metrics combination (ΔT, ΔFSN) is nearly 10%, whereas the FPR for the single metric ΔT exceeds 86% and the FPR for the ΔFSN is nearly 38%. Regarding the metric combination (ΔT, CRC), the FPR is 14.5%, while the FPR for the single metrics ΔT and CRC are 86.5% and 99.9%, respectively. In some other cases such as the metrics combinations (ΔT, NAV) and (ΔFSN, NAV), the FPR goes down to 10%. FPR of 14.5%, is obtained for the three metric combination of (ΔT, ΔFSN, CRC). However, for all other 3 metric combinations, FPR results are less than 5%.

TABLE II. FPR FOR SCENARIO 1 AND SCENARIO 2

Metrics		**Scenario 1**	**Scenario 2**
		FPR (%)	FPR (%)
1	ΔT	86.5	86.5
	ΔFSN	25.2	37.8
	NAV	2.3	18.5
	CRC	99.9	99.9
2	ΔT, ΔFSN	10.3	10.3
	ΔT, NAV	0.02	1.72
	ΔT, CRC	14.5	14.5
	ΔFSN, NAV	0	0
	ΔFSN, CRC	4.72	4.72
	NAV, CRC	4.72	4.95
3	ΔT, ΔFSN, NAV	1.94	3.31
	ΔT, ΔFSN, CRC	14.5	14.5

It is worth noting that the increase of the FPRs for the metrics combination including NAV is significantly smaller than the FPR results obtained with the single metric NAV in the scenario 2. Finally, we notice a clear improvement of the FPR of the metric combination compared to the FPR presented in [17]. Taking into account the multiple metric combinations which are common in both papers, we notice a reduction in FPRs for the metric combination of (ΔT, ΔFSN), reducing from 25% in [17] to 10.3%. Similarly, the FPR for (ΔT, NAV) is reduced from 86% in [17] to 1.72% (considering the worst case in scenario 2), and finally the FPR for (ΔT, ΔFSN, NAV) is reduced from 25% in [17] to 3.31%.

C. Performance Analysis Under Attack Traffic

Table III shows the performance evaluation in terms of DR, FPR, FNR, OSR, Precision and F-Measure for the attack scenarios listed in Table I (Scenarios 3-14). The results presented in Table III have been obtained using a sliding window size of 50 samples, which represents the case when the best performance was observed. The set of data is made up of 14 metric combinations and has been analysed for 12 scenarios. The results for the whole dataset are described by providing the median, the minimum and the maximum value of each evaluation parameter. In addition, the MAD (Mean Absolute Deviation) around the median is estimated, which is defined as $MAD = E[|X - median(X)|]$.

The results show that the proposed H-NIDS exhibits good performance for the single metric ΔT and NAV, as well as the metric combinations of (ΔT, NAV), (NAV, CRC) and (ΔFSN, NAV, CRC). In general, the hybrid solution produces a general performance improvement for the metric combinations when their performance is compared against the single metrics that are part of these combinations.

The single metric ΔT produces generally high DR results (81%), even though with a relatively high MAD value (14.5%). The minimum value of DR 0% shows that the metric could completely fail to detect the attack in some cases; this also accounts for the observed high MAD value. The FPR is about 59.9%, with a high MAD value. The FNR shows fair results as well; the median is 19% with a high MAD value of 14.5%. In general, the metric ΔT provides fair results, as indicated by the OSR, Precision and F-Measure, which range between low and high MAD values; 8.7%, 10.1% and 14.9% respectively. This low performance is caused because the legitimate clients are prevented from sending frames over the wireless medium during the attack and, consequently, the monitoring system cannot update the metrics for the calculation of the respective beliefs with enough frequency to strengthen the statistics of normal behaviour and allow detection of the attack instances.

The single metric ΔFSN provides a low DR (57.2%) with high FNR and FPR (42.8% and 40.7%), all of them characterised by a high MAD value. Although the OSR is quite high, the Precision is very low and characterised by a high MAD value. The ΔFSN metric, as explained above for the metric ΔT, is not frequently updated by the monitoring system during the attack. Fig. 4 shows the ΔFSN over time for Scenario 6. The majority of the normal instances (in blue) do not exceed 100 ΔFSN. However, around second 1 and second 70, there is a cluster of normal instances that deviate from the majority of normal data, which generates a large number of FPs. The red part of the graph represents the ΔFSN values of the attacker.

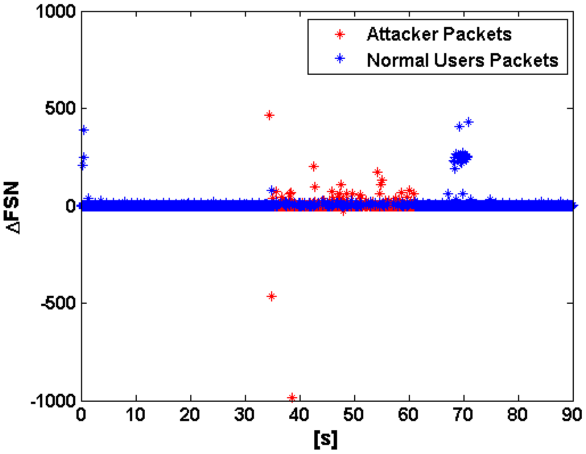

Figure 4. ΔFSN measurements in Scenario 6.

The single metric NAV provides excellent results with a perfect DR and low FNR and FPR. In every scenario, the DR is 100% (see also the MAD value, which is equal to zero). The FPR is about 4%, the minimum value for the FPR is about 0%

TABLE III. EXPERIMENTAL RESULTS

%		ΔT	ΔFSN	NAV	CRC	ΔFSN ΔT	NAV ΔT	CRC ΔT	NAV ΔFSN	CRC ΔFSN	CRC NAV	NAV ΔFSN ΔT	CRC ΔFSN ΔT	CRC NAV ΔFSN
DR	*Median*	81.0	57.2	100	100	7.6	80.7	21.2	57.2	29.6	100.0	38.7	24.5	70.8
	MAD	14.5	25.3	0	0	7.6	14.8	17.4	25.2	17.6	0.0	34.3	13.7	10.6
	Min	0	6.2	100	100	0.0	0.0	2.9	6.2	2.3	99.6	0.0	3.6	43.6
	Max	100	86.9	100	100	37.2	100.0	71.5	86.9	59.8	100.0	86.9	63.0	87.2
FPR	*Median*	59.9	40.7	3.8	87.1	2.3	0.0	6.9	0.0	11.3	4.7	0.2	7.1	7.2
	MAD	11.3	7.9	2.8	5.1	2.3	0.0	6.2	0.0	7.5	3.7	0.2	6.9	4.8
	Min	0	13.5	0.1	73.3	0.0	0.0	0.1	0.0	0.6	0.2	0.0	0.1	0.5
	Max	81.5	59.6	34.7	96.2	34.5	0.0	18.0	34.5	40.9	13.5	34.5	40.9	40.9
FNR	*Median*	19.0	42.8	0	0	92.4	19.3	78.8	42.8	70.4	0.0	61.3	75.5	29.2
	MAD	14.5	25.3	0	0	7.6	14.8	17.4	25.2	17.6	0.0	34.3	13.7	10.6
	Min	0	13.1	0	0	62.8	0.0	28.5	13.1	40.2	0.0	13.1	37.0	12.8
	Max	100	93.8	0	0	100.0	100.0	97.1	93.8	97.7	0.4	100.0	96.4	56.4
OSR	*Median*	36.7	52.1	96.2	12.9	84.6	96.5	83.4	90.7	76.7	95.3	88.7	81.6	89.6
	MAD	8.7	8.1	2.8	5.1	7.0	2.5	7.8	4.9	9.8	3.7	1.8	7.5	3.8
	Min	18.5	37.1	65.3	3.8	54.0	73.4	64.5	62.3	47.9	86.5	62.3	47.9	55.9
	Max	96.2	78.5	99.9	26.7	96.2	100.0	92.6	96.7	91.4	99.8	96.7	93.0	93.7
Prec.	*Median*	11.8	15.8	68.7	12.9	15.4	100.0	34.6	99.7	20.1	77.3	70.9	30.6	51.5
	MAD	10.1	9.7	16.0	5.1	13.3	0.0	17.6	0.3	11.9	18.9	29.1	15.7	21.5
	Min	0	0.6	20.8	3.8	0.0	0.0	4.2	30.4	4.1	30.1	0.0	14.8	25.9
	Max	25.7	29.3	99.6	26.7	34.9	100.0	85.3	100.0	77.9	99.0	100.0	84.9	92.4
F-score	*Median*	20.7	24.6	81.5	22.8	8.0	89.3	24.4	46.8	23.9	87.2	41.1	21.8	56.0
	MAD	14.9	11.9	11.6	7.6	8.0	8.4	15.4	9.7	9.2	12.1	23.9	7.7	10.7
	Min	0	1.1	34.4	7.3	0.0	0.0	3.5	11.6	3.9	46.3	0.0	7.0	38.5
	Max	40.9	43.8	99.8	42.1	33.3	100.0	59.0	93.0	53.5	99.5	93.0	53.5	87.5

and the worst FPR is about 34%. The MAD value for the FPR is very low (2.8%). The FNR is zero in all scenarios. On average the single metric NAV provides high OSR and high Precision.

Finally, the single metric CRC provides 100% DR, but with high FPR (87.1%). The single metric CRC, along with the single metric ΔFSN, provide the worst results among all the metrics. Fig. 5 shows the normalised CRC metric over time for Scenario 8. The CRC metric is very volatile and does not necessarily show a clear distinction between normal and attack instances, which can compromise the accuracy of the detector when used in isolation. However, when used along other metrics, such as (ΔFSN, ΔT), the DR improves from 7.6% to 24.5%.

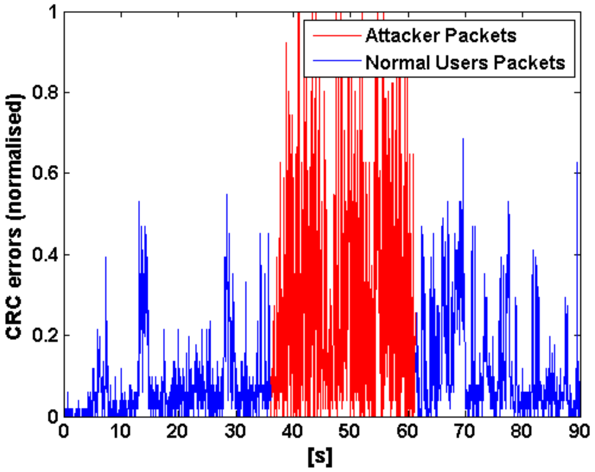

Figure 5. CRC errors measurements in Scenario 8.

Regarding the results generated by the two metric combinations, the combination (ΔT, NAV) is characterised by a high DR (80.7%) and FNR (19.3%), and a FPR equal to 0%. The low values of the MAD for the DR, FPR and FNR show that the aforementioned results are valid for the whole dataset. This metric combination generates high OSR (96.5%) and shows a clear improvement of the FPR (from 59.9% to 0%) when compared to the single metric ΔT. The OSR is also improved, when compared to ΔT.

The metric combination (ΔFSN, NAV) is characterised by fair DR (about 57.2%), FPR (about 0%) but high FNR (about 43%). In overall, this metric combination provides good results, with high OSR (90.7%) and Precision (99.7%). However, the high FNR, drops the performance of F-Score to 46.8%.

The metric combinations of (ΔT, ΔFSN), (ΔT, CRC), and (ΔFSN, CRC) provide all bad results. The DR and the FPR are quite low, but the FNR as well as the MAD values are high for these cases.

Among all possible two metric combination, the set (NAV, CRC) is the one that provides the best results. The performance of this set of metrics is similarly good to the results generated by the single metric NAV. In detail, the DR is 100%, the FPR is lower than 5% and the FNR is zero. These results are observed in all the scenarios, as confirmed by very low value of MAD (close to 0%) for the DR, the FNR and the FPR.

With reference to the performance of the single metric NAV shown in Table II, the metrics combination (NAV, CRC) has a very low FPR (4.7%). This is especially evident in the scenario 2, where the NAV is high not because of an attack is taking place but due to a legitimately high NAV value generated by a user. Therefore, the metric combination (NAV, CRC) not only provides excellent results, but also solves the problem related to the high FPR when the NAV is high for legitimate users. The very good results provided by (NAV, CRC) are confirmed also by the values of OSR, Precision and F-Score.

Regarding all the three metric combinations, the best results are provided by (ΔFSN, NAV, CRC), which produce high DR and low FPR. The MAD value shows that the achieved FPRs are generally low and that the high FNR on average makes the solution a fair solution. The fair performance of this metric combination is also confirmed by the high OSR.

VI. CONCLUSIONS

In this work, we have tackled the problem of identifying virtual jamming attacks on IEEE 802.11 networks. We proposed novel hybrid NIDS based on DS theory able to efficiently detect NAV attacks. This novel detector, which extends the method that we previously proposed in [17], takes advantage of the two types of IDSs. The high DR performance generally generated by signature-based NIDSs along with the ability to detect novel attacks provided by the anomaly-based NIDSs. The detection process involves the combination of beliefs from different metrics across multiple layers of observation in order to produce a collective decision on whether a NAV attack takes place or not. The beliefs are combined with the DS theory of evidence.

In order to evaluate the proposed solution, the hybrid NIDS has been tested on a real wireless scenario. A list of 14 different scenarios was proposed to emulate realistic scenarios in Wi-Fi networks. These scenarios include cases in which a client is located in a fixed position, keeping a constant distance to the AP, cases in which random movement is introduced to emulate an actual mobile behaviour, and mixed scenarios including both fixed and moving nodes to assess the performance of the detection algorithm within the same room when multiple clients are competing for the available wireless channel. The devices in all these scenarios used UDP traffic. Lastly, a test using TCP traffic was also carried out to study the effect of jamming attack on the establishment of a TCP connection.

The performance results of the proposed hybrid NIDS has been evaluated using six well-known parameters. These are DR, FPR, FNR, OSR, Precision and F-Score. We have evaluate the performance results generated when different metrics combination are used, as well as single metrics. Among all the single metrics, the solution that exhibits the best results is NAV, which generates 100% DR, 3.8% of FPR and 0% FNR. As for the different metrics combination, the set (ΔT, NAV), (NAV, CRC) and (ΔFSN, NAV, CRC) generate performance results as good as the single metric NAV. These metrics combination generate good results for several of the tested real Wi-Fi scenarios. Overall, the results evidenced by the hybrid NIDS outperforms the detection results generated by the anomaly-based NIDS presented in [17].

As for future work, we will focus our work on the development of a real-time hybrid NIDS able to detect a wider range of threats and cyber-attacks against wireless networks. Similarly, we will extend the implementation of the proposed hybrid NIDS to other wireless communication technologies, such as LTE and WiMAX. In addition, we wish to add the capability of automatic selection of relevant metrics tailored to specific types of attacks.

REFERENCES

[1] N. Nostro, A. Ceccarelli, A. Bondavalli, and F. Brancati, “A methodology and supporting techniques for the quantitative assessment of insider threats,” Proc. of the 2nd Int. Workshop on Dependability Issues in Cloud Computing (DISCCO), vol. 3, pp. 1-6, 2013.

[2] N. Nostro, A. Ceccarelli, A. Bondavalli, and F. Brancati, “Insider threat assessment: A model-based methodology,” in ACM SIGOPS Operating Systems Review, vol. 48, no. 2, pp. 3-12, 2014.

[3] SESP Group, “SESP RF jammers,” Available: http://www.sesp.com (Access Date: 3 March, 2017).

[4] Phonejammer, “Mobiledevice jammer,” Available: http://www.phonejammer.com/ (Access Date: 3 March, 2017).

[5] Ettus Research, “Software Defined Radios (SDR),” Available: http://www.ettus.com/home (Access Date: 3 March, 2017).

[6] A. Mahanti, N. Carlsson, C. Williamson, and M. Arlitt, “Ambient interference effects in Wi-Fi networks,” Proc. of the 9th International IFIP TC 6 Conference on Networking (NETWORKING), Lecture Notes in Computer Science, vol. 6091, pp. 160-173, 2010.

[7] M. Raya, I. Aad, J.-P. Hubaux, and A. El Fawal, “DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 hotspots,” in IEEE Transactions on Mobile Computing, vol. 5, no. 12, pp. 1691-1705, 2006.

[8] L. Montecchi, N. Nostro, A. Ceccarelli, G. Vella, A. Caruso, and A. Bondavalli, “Model-based evaluation of scalability and security tradeoffs: A case study on a multi-service platform,” in Electronic Notes in Theoretical Computer Science, vol. 310, pp. 113-133, 2015.

[9] G. Thamilarasu, S. Mishra, and R. Sridhar, “A cross-layer approach to detect jamming attacks in wireless ad hoc networks,” Proc. of the Military Communications Conference (MILCOM), pp. 1-7, 2006.

[10] L. Wang, and A. M. Wyglinski, “A combined approach for distinguishing different types of jamming attacks against wireless networks,” Proc. of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim), pp. 809-814, 2011.

[11] A. G. Fragkiadakis, V. A. Siris, N. E. Petroulakis, and A. P. Traganitis, “Anomaly-based intrusion detection of jamming attacks, local versus collaborative detection,” in Wireless Communications and Mobile Computing, vol. 15, no. 2, pp. 276-294, 2015.

[12] D. Chen, J. Deng, and P. K. Varshney, “Protecting wireless networks against a denial of service attack based on virtual jamming,” Proc. of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom), pp. 1-2, 2003.

[13] P . García-Teodoro, J. Díaz-V erdejo, G. Maciá-Fernández, and E. Vázquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges,” in Computers & Security, vol. 28, no. 1, pp. 18-28, 2009.

[14] D.-R. Tsai, W.-P. Tai, and C.-F. Chang, “A hybrid intelligent intrusion detection system to recognize novel attacks,” Proc. of the IEEE 37th Annual International Carnahan Conference on Security Technology, pp. 428-434, 2003.

[15] T.-S. Chou, and T.-N. Chou, “Hybrid classifier systems for intrusion detection,” Proc. of the 7th Annual Communication Networks and Services Research Conference (CNSR), pp. 286-291, 2009.

[16] J. Chen, and D. Yang, “Intrusion detection system platform based on light-weighted hybrid artificial immune algorithms,” Proc. of the 5th International Conference on Natural Computation (ICNC), pp. 319-324, 2009.

[17] G. Escudero-Andreu, K. G. Kyriakopoulos, F. J. Aparicio-Navarro, D. J. Parish, D. Santoro, and M. Vadursi, “A data fusion technique to detect wireless network virtual jamming attacks,” Proc. of the IEEE International Workshop on Measurements & Networking (M&N), pp. 1-6, 2015.

[18] G. Shafer, “A mathematical theory of evidence”, Princeton University Press, 1976.

[19] NS-3 Network Simulator, Available: https://www.nsnam.org/ (Access

Date: 3 March, 2017).

[20] X. Zeng, R. Bagrodia, and M. Gerla, “GloMoSim: A library for parallel simulation of large-scale wireless networks,” Proc. of the Parallel and Distributed Simulation (PADS), pp. 154-161, 1998.

[21] F. J. Aparicio-Navarro, K. G. Kyriakopoulos, and D. J. Parish, “A multi-layer data fusion system for Wi-Fi attack detection using automatic belief assignment,” Proc. of the World Congress on Internet Security (WorldCIS), pp. 45-50, 2012.

[22] C. N. Modi, and D. Patel, “A novel hybrid-network intrusion detection system (H-NIDS) in cloud computing,” Proc. of the IEEE Symposium on Computational Intelligence in Cyber Security (CICS), pp. 23-30, 2013.

[23] Cisco, “Snort Network Intrusion Detection & Prevention System,” Available: https://www.snort.org/ (Access Date: 3 March, 2017).

[24] R. R. Karthick, V. P. Hattiwale, and B. Ravindran, “Adaptive network intrusion detection system using a hybrid approach,” Proc. of the Fourth International Conference on Communication Systems and Networks (COMSNETS), pp. 1-7, 2012.

[25] X. Cheng, and S. Wen, “A real-time hybrid intrusion detection system based on principle component analysis and self-organizing maps,” Proc. of the Sixth International Conference on Natural Computation (ICNC), pp. 1182-1185, 2010.

[26] O. Galinina, A. Pyattaev, S. Andreev, M. Dohler, and Y. Koucheryavy, “5G Multi-RAT LTE-WiFi ultra-dense small cells: Performance dynamics, architecture, and trends,” in IEEE Journal on Selected Areas in Communications, vol. 33, no. 6, 1224-1240, 2015.

[27] D. Yu, and D. Frincke, “Alert confidence fusion in intrusion detection systems with extended Dempster-Shafer theory,” Proc. of the 43rd Annual Southeast Regional Conference, vol. 2, pp. 142-147, 2005.

[28] A. Ferrero, and S. Salicone, “Uncertainty: Only one mathematical approach to its evaluation and expression?,” in IEEE Transactions on Instrumentation and Measurement, vol. 61, no. 8, pp. 2167-2178, 2012.

[29] S. Salicone, Measurement uncertainty: An approach via the mathematical theory of evidence, Springer Series in Reliability Engineering, 2007.

[30] A. Ferrero, R. Gamba, and S. Salicone, “A method based on random-fuzzy variables for the on-line estimation of the measurement uncertainty of DSP-based instruments,” in IEEE Transactions on Instrumentation and Measurement, vol. 53, no. 5, pp. 1362-1369, 2004.

[31] I. Ruthven, and M. Lalmas, “Using Dempster-Shafer's theory of evidence to combine aspects of information use,” in Journal of Intelligent Information Systems, vol. 19, no. 3, pp. 267-301, 2002.

[32] G. Combs, “Wireshark-network protocol analyser,” Available: https://www.wireshark.org/ (Access Date: 3 March, 2017).

[33] NLANR/DAST “iPerf – The TCP, UDP and SCTP network bandwidth measurement tool,” Available: https://iperf.fr/ (Access Date: 3 March, 2017).

[34] A. Y. Dak, N. E. A. Khalid, and S. Yahya, “A novel framework for jamming detection and classification in wireless networks,” Proc. of the 8th International Conference on Computing and Networking Technology (ICCNT), pp. 240-246, 2012.

[35] Gast, Matthew. 802.11 wireless networks: the definitive guide. " O'Reilly Media, Inc.", 2005.

[36] Gao, Deyun, Jianfei Cai, and Chuan Heng Foh. "Medium access cooperations for improving VoIP capacity over hybrid 802.16/802.11 cognitive radio networks." International Conference on Research in Networking. Springer Berlin Heidelberg, 2008.